

IMPLEMENTASI METODE ASYNCHRONOUS JAVASCRIPT AND XML (AJAX) PADA PEMBUATAN WEBSITE UNIVERSITAS SAM RATULANGI

Hendra Jindry Lamani, Hans Wowor, Arthur Rumagit, Nancy Tuturoong

Abstrak—Dalam era teknologi modern saat ini, informasi yang dapat diakses secara cepat dan akurat menjadi salah satu kebutuhan penting. Berbagai media digunakan untuk mendapatkan informasi diantaranya situs Universitas Sam Ratulangi Manado yang didalamnya berisikan informasi tentang universitas ini sebagai salah satu lembaga pendidikan tinggi terbesar di Sulawesi Utara.

Situs Universitas Sam Ratulangi Manado yang dibuat dengan menggunakan teknologi *web* konvensional, lambat dalam pemuatan halaman baru (*loading*). Ini terjadi karena pada saat pemuatan halaman baru seluruh isi halaman baru harus dimuat kembali meskipun perbedaan isi dengan halaman sebelumnya hanyalah sedikit. Untuk itu dibuat Situs Universitas Sam Ratulangi Manado baru yang menggunakan AJAX yang memiliki fitur – fitur interaktif dan waktu tunggu kecil. Hasilnya situs yang dibangun dengan teknologi AJAX waktu pemuatan halaman baru (*loading page*) menjadi lebih cepat.

Indeks - AJAX, asynchronous, synchronous, JavaScript

I. PENDAHULUAN

Tidak dapat dipungkiri bahwa pengguna dunia *internet* selalu menghendaki akses yang cepat ketika sedang membuka halaman – halaman dari satu situs. Disamping itu pula tampilan situs yang menyerupai tampilan *desktop* saat pergantian halaman baru menjadi hal yang memberikan daya tarik tersendiri bagi para pengunjung. Sementara itu diyakini bahwa pembuatan situs dengan menggunakan CMS sangat memberikan kemudahan – kemudahan namun terbatas dalam hal – hal tertentu.

Situs Universitas Sam Ratulangi Manado yang dibuat dengan menggunakan teknologi *web* tradisional, lambat dalam pemuatan halaman baru (*loading*). Ini terjadi karena pada saat pemuatan halaman baru seluruh isi halaman baru harus dimuat kembali meskipun perbedaan isi dengan halaman sebelumnya hanyalah sedikit. Apabila koneksi *internet* lambat maka pada layar monitor akan nampak warna putih sebagai tanda halaman baru sementara dalam proses pemuatan. Proses demikian dikenal dengan nama *synchronous*. Akan lebih baik tentunya jika pemuatan halaman baru situs Universitas Sam Ratulangi Manado dapat terjadi secara lebih cepat dan warna putih pada layar monitor dapat dihindari.

Sejak diperkenalkan, teknologi AJAX telah memperlihatkan kelebihan – kelebihannya dibandingkan dengan teknologi pembuatan *web* tradisional sebelumnya. Karena kehebatannya teknologi AJAX saat ini telah diimplementasikan oleh google dalam pengembangan situs gmail bahkan dalam pengembangan situs – situs terkenal dan besar lainnya.

II. DASAR TEORI

Teknologi AJAX (*Asynchronous javascript and XML*) diperkenalkan oleh Jesse James Garret dari Adaptive Path tahun 2005. Ia mendiskripsikan bagaimana mengembangkan *Web* yang berbeda dengan metode tradisional melalui artikelnya yang berjudul “*Ajax : A new Approach to Web Applications*”. Dalam artikel ini ia yakin bahwa aplikasi *Web* dapat menutup jurang pemisah antara *Web* dan aplikasi desktop (Andi Sunyoto, MKom, 2007).

Andi Sunyoto, MKom (2007) lebih jauh menuliskan bahwa pengembangan *Web* secara tradisional bekerja secara *synchronously* antara aplikasi dan *server*, setiap kali melakukan *link* atau melakukan operasi *submit* pada *form*. Caranya *browser* mengirim data ke *server*, *server* merespons dan seluruh halaman akan di *refresh*.

Aplikasi *Web* yang bekerja dengan AJAX bekerja secara *Asynchronously* yang berarti mengirim dan menerima data dari *user* ke *server* tanpa perlu memuat kembali seluruh halaman, melainkan hanya melakukan pergantian pada bagian *Web* yang hendak diubah. Penggunaan AJAX mulai populer ketika digunakan oleh Google pada tahun 2005.

AJAX bukanlah bahasa pemrograman baru, tetapi merupakan teknik baru penggunaan standart yang telah ada. Dengan AJAX kita menjadi lebih baik, cepat dan menambah *user-friendly* dan interaktif pada aplikasi *Web* kita. AJAX berbasiskan pada *Javascript* dan request HTTP (<http://www.w3schools.com>, 2007). AJAX berbasiskan standar terbuka seperti :

- *Javascript*
- XML
- HTML / XHTML
- CSS

Lebih jauh dalam <http://www.w3schools.com> dituliskan bahwa dengan menggunakan *Javascript* AJAX dapat mengirim dan menerima data antara *Web server* dan *Web browser*. Teknik yang dimiliki AJAX akan bergantian tukar data dan memuat kembali seluruh halaman. Melalui AJAX, *Javascript* dapat dikomunikasikan langsung dengan *server* menggunakan objek *Javascript XMLHttpRequest*. Objek *Javascript* ini dapat menampilkan data dari *Web server* tanpa harus memuat kembali seluruh halaman.

AJAX menggunakan *Asynchronous* data transfer (pada HTTP request) antara *browser* dan *Web server*, yang memperbolehkan halaman *Web* me-request bit yang kecil atau seluruh informasi dari *server*. Teknik AJAX membuat aplikasi *internet* menjadi kecil, cepat dan lebih *user-friendly*. AJAX adalah aplikasi *Web* yang lebih baik dan menambah keuntungan dibanding aplikasi desktop seperti dapat menjangkau pengguna yang luas, mudah diinstal, mudah dikembangkan dan mudah dipelihara (Raymond, Scott, 2006).

III. METODOLOGI PENELITIAN

A. Objek Penelitian

Penelitian ini dilakukan di Universitas Sam Ratulangi Manado. Penelitian dengan judul 'Implementasi Metode *Asynchronous JavaScript* and XML (AJAX) pada Pembuatan *Website* Universitas Sam Ratulangi' dalam pembuatannya dipisahkan atas sisi administrator dan sisi pengguna (*user*).

B. Alat dan Bahan Penelitian

Website berbasis AJAX yang dibuat ini masih menggunakan *localhost* sebagai *server*nya. Itu berarti spesifikasi perangkat keras yang dibutuhkan untuk *server* tidak harus seperti yang digunakan oleh *server* untuk *business web hosting*. Komputer dengan processor Pentium IV dan memory minimal 512 MB sudah dapat digunakan sebagai *server (localhost)* untuk mendemonstrasikan *Website* ini.

Sementara itu pada sisi *client* semakin tinggi spesifikasi *hardware* yang dimiliki akan semakin baik dalam mengakses *Website* berbasis AJAX ini.

Perangkat lunak yang dibutuhkan pada sisi *server* berupa *web server*, *php engine* dan *database server*. Untuk *web server* digunakan *Apache web server* dan *MySQL* digunakan sebagai *database server*. Untuk simulasi *Website* yang dibuat ini digunakan *WAMP SERVER* versi 2.1 yang didalamnya sudah terdapat *Apache* versi 2.2.17, *PHP* versi 5.3.5 dan *MySQL* versi 5.5.8

C. Tinjauan Sistem Yang Sedang Berjalan

Tinjauan terhadap *Website* Universitas Sam Ratulangi Manado (UNSRAT) dengan alamat www.unsrat.ac.id menunjukkan kelebihan *Website* dari sisi *content* yang menyediakan fasilitas Portal Akademik, *webmail*, dan daftar link untuk e-jurnal ilmiah.

Namun ada juga kelemahan dari *Website* UNSRAT dari sisi kecepatan pergantian halaman. Setelah kita masuk ke dalam *Website* UNSRAT kemudian mengklik salah satu menu (*link*) maka seluruh halaman akan dimuat kembali padahal

perbedaan isi halaman baru dengan halaman sebelumnya hanya sedikit. Hal ini tentunya akan berdampak pada kecepatan akses terhadap satu halaman apalagi pada waktu – waktu sibuk.

Pengujian menggunakan aplikasi pengujian pada *Website* UNSRAT diperoleh data yang menunjukkan bahwa rata-rata waktu tanggap (*respon time*) untuk pergantian halaman berkisar 19s dengan ukuran file berkisar 1.7 MB dan *throughput* 124 *request*. Dengan waktu tanggap yang relatif lama tersebut membuat pengunjung *Website* merasa kurang nyaman.

Dalam interaksi manusia dan komputer salah satu faktor yang harus dipenuhi adalah faktor perancangan tampilan antar muka. Sistem yang dibangun harus mampu menarik perhatian dan menyenangkan saat digunakan. Tampilan halaman *Website* dari *Website* UNSRAT yang ada saat ini terasa kurang menarik, karena penggunaan paduan warna pada halaman *Website* yang monoton.

D. Pemodelan Sistem Administrator

Pemodelan pada sisi admin dilakukan dengan pendekatan metode terstruktur menggunakan *tool* pemodelan pendekatan terstruktur yaitu Diagram Aliran Data (DAD). Diagram konteks sisi administrator bias dilihat pada gambar 1

Dua buah *entity* berhubungan langsung dengan sistem administrator yaitu *entity* admin dan *entity* operator yang dapat diuraikan sebagai berikut :

1. Administrator

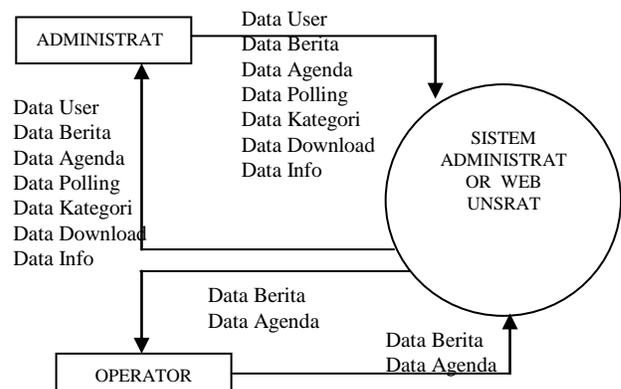
Input : Data *User*, Data Berita, Data Agenda, Data Polling, Data Kategori, Data *Download*, Data Info

Output : Data *User*, Data Berita, Data Agenda, Data Polling, Data Kategori, Data *Download*, Data Info

2. Operator

Input : Data Berita, Data Agenda

Output : Data Berita, Data Agenda



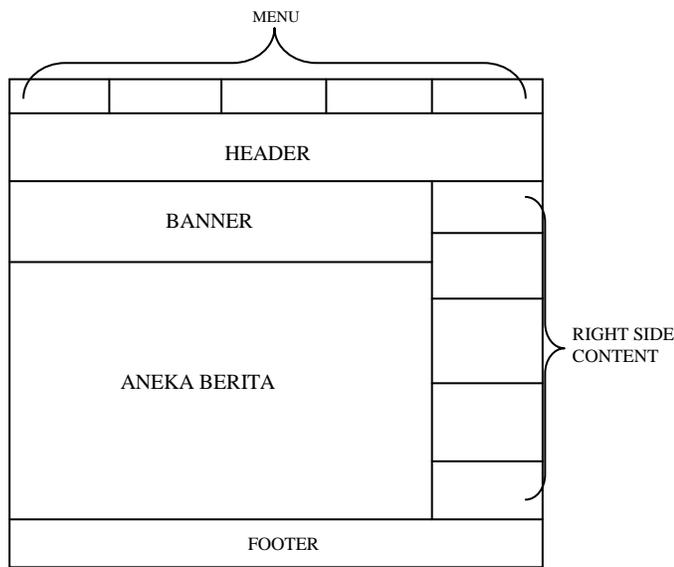
Gambar 1. Diagram konteks sisi administrator

E. Layout Situs

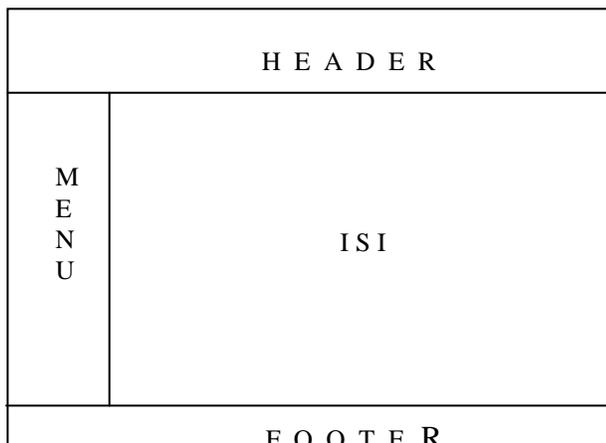
Untuk membuat *layout website* digunakan konsep CSS *layout*. *Skrrip CSS* diembedded ke kode html dimana kode html bertanggung jawab hanya dengan struktur situs sedangkan *style* situs diatur oleh kode CSS. Untuk pembuatan menu pada situs ini selain html dan CSS digunakan juga *client side script* dalam hal ini *JavaScript*.

Agar situs ini interaktif, dinamis dengan waktu pemuatan halaman baru lebih cepat maka diaplikasikan teknologi AJAX dalam pembuatannya. Pada gambar *layout* sisi *client* sebagaimana ditampilkan pada gambar 2, bagian *menu*, *header*, *banner*, *footer* dan *right content* adalah bagian yang statis dan hampir selalu ada pada setiap pergantian halaman baru. Bagian aneka berita dirancang sebagai tempat menampilkan informasi yang diminta *user*. Singkatnya pada setiap pergantian halaman baru hanya bagian aneka berita yang berubah.

Sisi admin dibuat sehingga memungkinkan administrator dapat mengelola isi situs berbasis *content management system*. Proses *input*, *delete* dan *update* berita serta *entity – entity* lainnya dilakukan di *interface* ini. Layout website sisi administrator bias dilihat pada gambar 3.



Gambar 2. Layout website sisi client



Gambar 3. Layout website sisi admin

IV. HASIL

A. Implementasi Antar Muka

Halaman utama sisi *client* adalah halaman yang pertama kali dikunjungi *client* ketika mengakses website ini. Didalam halaman utama ini terdapat beberapa menu, berita – berita, agenda dan item – item web lainnya. Pengunjung bisa pindah ke halaman lain dengan cara *mengklik link* yang tersedia. Tampilan halaman utama sisi *client* bias dilihat paa gambar 4.

Halaman utama administrator ditampilkan apabila pengguna berhasil *login* dengan memasukan *username* dan *password*. Didalamnya berisikan menu – menu untuk mengelola isi dari website unsrat. Tampilan halaman utama sisi administrator bias dilihat pada gambar 5.

B. Perbandingan Dengan Website Resmi UNSRAT

Situs yang dibuat memenuhi standart interaksi manusia dan komputer dimana, sistem berfungsi sesuai yang diinginkan (*useful*), sistem mudah dioperasikan (*usable*), sistem memotivasi penggunaanya untuk menggunakannya, menarik dan menyenangkan (*used*).



Gambar 4. Halaman utama sisi client



Gambar 5. Halaman utama sisi administrator

Agar website yang dibangun interaktif, dinamis dengan waktu pemuatan halaman baru lebih cepat maka diaplikasikan teknologi AJAX dalam pembuatannya. Halaman utama pada sisi *client* khususnya bagian menu, *header*, *banner*, *footer* dan *right content* adalah bagian yang statis dan hampir selalu ada pada setiap pergantian halaman baru. Bagian aneka berita dirancang sebagai tempat menampilkan informasi berita yang diminta *user*. Apabila *user* mengklik salah satu *link* yang ada maka halaman baru akan dimuat, namun dalam *project* yang dibuat ini dengan mengimplementasikan teknologi AJAX maka hanya isi yang baru dari satu halaman baru yang akan dimuat sementara isi yang tidak berubah tidak akan dimuat lagi.

Berbeda dengan *website* UNSRAT yang sekarang yang belum menggunakan teknologi AJAX, pada setiap pergantian halaman baru seluruh isi halaman akan dimuat kembali. Hal ini tentunya akan berdampak pada kecepatan akses terhadap satu halaman apalagi pada waktu – waktu sibuk.

Pada situs yang dibuat ini, saat *user* mengirimkan *httprequest* maka request tidak langsung diteruskan ke web server melainkan ke *ajax engine* yang nantinya akan mengkomunikasikan request tadi ke web server. Demikian juga response dari web server tidak langsung ditampilkan oleh web *browser*, melainkan oleh *AJAX engine* terlebih dahulu akan dilakukan parsing data dan perubahan – perubahan sedikit terhadap response tersebut untuk keperluan menampilkan informasi pada web *browser*.

AJAX engine diimplementasikan melalui dua fungsi berikut ini :

```
function loadXMLDoc(url,cfunc)
{
    if (window.XMLHttpRequest)
    {
        xmlhttp=new XMLHttpRequest();
    }
    else
    {
        xmlhttp=newActiveXObject
        ("Microsoft.XMLHTTP");
    }

    xmlhttp.onreadystatechange=cfunc;
    xmlhttp.open("GET",url,true);
    xmlhttp.send();
}

function displayCari(str,no)
{
    loadXMLDoc ("displaycari.php?id="+str+"&urut="+
    no, function()
    {
        if(xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            document.getElementById("ajaxKiri").innerHTML+xmlhttp.responseText;
        }
    });
}
```

Awalnya ketika satu *event ditrigger* pada web *browser* seperti pada kode berikut `` maka fungsi *displayCari* yang memiliki dua parameter akan dipanggil. Fungsi *displayCari* sendiri akan memanggil fungsi

loadXMLDoc() dimana fungsi ini akan membentuk *XMLHttpRequest* objek jika web *browser* yang digunakan adalah salah satu dari IE7, safari, chrome, atau opera dan akan membentuk *ActiveXObject* jika web *browser*nya adalah IE6 atau IE5.

XMLHttpRequest objek memiliki beberapa method diantaranya method *open()* dengan tiga parameter dan method *send ()*. Method – method inilah yang membuka dan mengirim request ke web server. Dalam hal ini setiap request dari web *browser* akan ditangani oleh halaman *displayCari.php* yang mempunyai dua parameter yaitu parameter *id* dan *urut*.

Response yang dikirim oleh web server akan ditangani kode javascript dan meletakkan hasil manipulasi response pada halaman html (pada bagian elemen *div* dengan attribute *id* ‘*ajaxKiri*’). Dengan demikian pada setiap request hanya isi dari elemen *div* dengan attribute *id* ‘*ajaxKiri*’ yang berubah sementara isi dari elemen lainnya tetap. Singkatnya pada setiap request yang terjadi hanyalah pertukaran data bukan pertukaran halaman seperti pada web UNSRAT yang sekarang.

Sisi admin pada situs yang dibuat untuk tugas akhir ini memungkinkan administrator dapat mengelola isi situs berbasis content management sistem. Proses *input*, *delete* dan *update* berita serta *entity – entity* lainnya dilakukan di *interface* ini. Agar dinamis, kode php diintegrasikan dengan database yang sebelumnya telah dibuat dengan menggunakan DBMS MySQL.

C. Pengujian

Nilai *total file (size)* pada web UNSRAT resmi berubah-ubah sesuai dengan aktivitas yang dilakukan dimana seluruh *file* dimuat ulang saat melakukan pergantian halaman. Nilai *total file (size)* pada web UNSRAT dengan AJAX bertambah saat melakukan aktivitas, ini disebabkan karena selama web diakses tidak terjadi pergantian halaman, namun terjadi pemuatan informasi baru, dan perubahan tampilan banner serta tampilan sekilas info. Perbandingan *total file (size)* antara web UNSRAT resmi dan web UNSRAT dengan AJAX bias dilihat pada tabel 1.

TABEL 1. PERBANDINGAN TOTAL FILE (SIZE)

| Aktivitas | Total file (size) | |
|--|-------------------|------------------------|
| | Web UNSRAT Resmi | Web UNSRAT Dengan AJAX |
| Load halaman utama website | 1.8 MB | 594.9 KB |
| Load halaman menu Fakultas (Fakultas Teknik) | 1.8 MB | 635.7 KB |
| Load halaman menu Kemahasiswaan (Organisasi Mahasiswa) | 1.7 MB | 708.4 KB |

TABEL 2. PERBANDINGAN *THROUGHPUT*

| Aktivitas | <i>Throughput</i> | |
|--|-------------------|------------------------|
| | Web UNSRAT Resmi | Web UNSRAT Dengan AJAX |
| Load halaman utama website | 134 request | 58 request |
| Load halaman menu Fakultas (Fakultas Teknik) | 124 request | 86 request |
| Load halaman menu Kemahasiswaan (Organisasi Mahasiswa) | 124 request | 177 request |

TABEL 3. PERBANDINGAN *RESPONSE TIME*

| Aktivitas | <i>Response Time (Onload)</i> | |
|--|-------------------------------|------------------------|
| | Web UNSRAT Resmi | Web UNSRAT Dengan AJAX |
| Load halaman utama website | 19.26 s | 7.49 s |
| Load halaman menu Fakultas (Fakultas Teknik) | 20.52 s | 7.49 s |
| Load halaman menu Kemahasiswaan (Organisasi Mahasiswa) | 19.87 s | 7.49 s |

Nilai *Throughput* pada web UNSRAT resmi berubah-ubah sesuai dengan aktivitas yang dilakukan dimana seluruh *file* dimuat ulang saat melakukan pergantian halaman. Nilai *Throughput* pada web UNSRAT dengan AJAX bertambah saat melakukan aktivitas, ini disebabkan karena selama web diakses tidak terjadi pergantian halaman, namun terjadi pemuatan informasi baru, dan perubahan tampilan banner serta tampilan sekilas info. Perbandingan nilai *throughput* antara web UNSRAT resmi dan web UNSRAT dengan AJAX bias dilihat pada tabel 2.

Nilai *Response Time (Onload)* pada web UNSRAT resmi berubah-ubah sesuai dengan aktivitas yang dilakukan dimana seluruh *file* dimuat ulang saat melakukan pergantian halaman. Nilai *Response Time (Onload)* pada web UNSRAT dengan AJAX bernilai tetap (7.49s) karena saat mengklik salah satu *link* (menu) tidak terjadi pergantian halaman hanya pemuatan informasi baru yang dimuat pada halaman yang sama. Perbandingan nilai *response time* antara web UNSRAT resmi dan web UNSRAT dengan AJAX bisa dilihat pada tabel 3.

Halaman web tidak dimuat ulang, hanya informasi yang di minta saja yang diperbarui. Pada tabel 4 bisa dilihat nilai total waktu *request*, nilai Total Ukuran (*zise*) dan nilai *Throughput* bertambah karena selama web diakses, tampilan banner dan tampilan info yang dibuat selalu berubah dengan periode waktu yang tetap. Sementara nilai *Response Time (Onload)* tetap karena saat mengklik menu Organisasi Mahasiswa, tidak terjadi pergantian halaman.

TABEL 4. PERBANDINGAN TOTAL WAKTU *REQUEST*

| Aktivitas | Total Waktu <i>Request</i> | |
|--|----------------------------|------------------------|
| | Web UNSRAT Resmi | Web UNSRAT Dengan AJAX |
| Load halaman utama website | 28.46 s | 10.69 s |
| Load halaman menu Fakultas (Fakultas Teknik) | 29.91 s | 20.24 s |
| Load halaman menu Kemahasiswaan (Organisasi Mahasiswa) | 29.16 s | 46.78 s |

V. PEMBAHASAN

Dari tabel perbandingan di atas bisa dilihat bahwa pada web UNSRAT dengan AJAX, nilai Total Waktu *Request*, nilai Total Ukuran (*zise*) dan nilai *Throughput* bertambah saat melakukan aktivitas, ini disebabkan karena selama web diakses tidak terjadi pergantian halaman, namun terjadi pemuatan informasi baru, dan perubahan tampilan banner serta tampilan sekilas info. *Response Time (Onload)* pada web UNSRAT dengan AJAX bernilai tetap (7.49 s) karena saat mengklik salah satu *link* (menu) tidak terjadi pergantian halaman hanya pemuatan informasi baru yang dimuat pada halaman yang sama.

Dari tabel perbandingan di atas dapat dilihat bahwa pada web resmi UNSRAT tanpa AJAX nilai Total Waktu *Request*, nilai Total Ukuran (*zise*) nilai *Throughput* dan nilai *Response Time (Onload)* selalu berubah, ini disebabkan setiap mengklik salah satu *link* (menu) maka seluruh isi halaman akan dimuat ulang. Ini tentunya akan berdampak pada kecepatan akses terhadap satu halaman apalagi pada waktu – waktu sibuk.

Pada situs yang dibuat ini, saat *user* mengirimkan *httprequest* maka *request* tidak langsung diteruskan ke *web server* melainkan ke *ajax engine* yang nantinya akan mengkomunikasikan *request* tadi ke *web server*. Demikian juga *response* dari *web server* tidak langsung ditampilkan oleh *web browser*, melainkan oleh *AJAX engine* terlebih dahulu akan dilakukan *parsing* data dan perubahan – perubahan sedikit terhadap *response* tersebut untuk keperluan menampilkan informasi pada *web browser*.

VI. KESIMPULAN

Situs yang dibangun dengan menerapkan teknologi AJAX memiliki kelebihan dibanding dengan situs yang dibangun dengan teknologi atau model tradisional. Kelebihan yang paling menonjol adalah bahwa situs yang dibangun dengan teknologi AJAX waktu pemuatan halaman baru (*loading page*) lebih cepat. Ini disebabkan karena pada situs AJAX yang dimuat bukanlah halaman melainkan data yang baru pada halaman situs yang diminta. Fitur – fitur handal yang ada pada JavaScript, teknologi XML dan teknologi

pembuatan *web* dinamis dipadukan dalam penerapan pembuatan situs berbasis teknologi AJAX.

Adapun hal – hal yang telah berhasil dicapai dalam penelitian ini adalah :

1. Halaman administrator dibuat berbasis *Content Management System* (CMS) sehingga pengelolaan atas isi *web* yang meliputi penambahan, *update* dan hapus data dapat dilakukan secara dinamis
2. Berita atau informasi yang ditampilkan pada sisi *client* telah menggunakan AJAX teknologi dalam pemuatan / pengaksesannya. Hal ini berdampak pada kecepatan akses untuk halaman tertentu.
3. Fitur – fitur penting dalam sebuah *website* meliputi *search*, *polling*, *download*, *banner* dll telah dibuat dalam *website* ini.

Apa yang telah dicapai diatas memberi dampak pada pengelolaan isi web menjadi lebih cepat dan mudah serta mengakses *website* ini menjadi lebih cepat.

REFERENCES

- [1] Al-Bahra. *Analisa dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu. 2005.
- [2] H. Lukamanul. 2011. *Bikin Website Super Keren dengan PHP & jQuery*. Yogyakarta: Lokomedia.
- [3] K. Abdul. *Dasar Pemrograman Web Dinamis Menggunakan PHP*. Yogyakarta: Penerbit Andi. 2005.
- [4] K. Abdul. *Zero to a PRO Membuat Aplikasi Web dengan PHP + Database MySQL*. Yogyakarta: Penerbit Andi. 2009.
- [5] K. Abdul. *Mastering Ajax dan PHP*. Yogyakarta: Penerbit Andi. 2009.
- [6] L. Craig. *Applying UML and Patterns*. USA: Pearson Education International. 2004.
- [7] N. Adi, *Analisis dan Perancangan Sistem Informasi dengan Metodologi Prosedural*. Bandung: Informatika. 2005.
- [8] S. Insap. *Interaksi Manusia dan Komputer*. Yogyakarta: Penerbit Andi. 2009.
- [9] S. Andi. *AJAX Membangun Web dengan Teknologi ASYNCHRONOUS JAVASCRIPT & XML*. Yogyakarta: Penerbit Andi. 2007.